IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol-6, Special Issue-4, May 2018 E-ISSN: 2347-2693

A Signal Integrity Based Rate Adaptation Algorithm

Mohan Kumar Seerangarajan

Senior System Engineer, Qualcomm, Sandiego, CA, USA

*Corresponding Author: arshree_23@yahoo.co.in

Available online at: www.ijcseonline.org

Abstract— Wireless systems have gained momentum in recent years due to ease, universality and cost reduction. In a wireless network the signal degradation is time variant based on environmental factors. As variations in signal quality and interference from external sources affect the network, it is necessary and important to adapt the rate to the channel conditions via a rate adaptation algorithm driven by changes in the signal quality. This research addresses signal integrity in a local area network and proposes a rate adaptation algorithm for 802.11 wireless networks based on signal quality and packet error rate. The relationship between signal strength and packet error rate is analyzed and a new rate adaptation algorithm for 802.11n is presented. The algorithm developed in this thesis effectively adapts the rate to minimize packet error rate while it maximizes throughput. Through experiments, it has been shown that the proposed algorithm achieves better throughput compared to state of the technology rate adaptation algorithms currently implemented in commercial 802.11n systems.

Keywords— Wireless, 802.11, MAC

I. INTRODUCTION

802.11n is the recent IEEE standard that governs wireless networking. There are various features of 802.11n which makes it ahead of the previous standards; one of them is MIMO and high rate transmission. The other advantage of 802.11n is that it can operate on both 2.4 and 5 GHz bands. The WLAN standard, IEEE 802.11 provides the multi-rate capability to balance the tradeoff between communication efficiency and reliability. For example, high transmission rate is adopted in channels of good quality and low rate is used in channel of poor quality. When operating in error-prone channels, lower data rates can overcome the channel conditions to achieve more reliable transmission than the higher data rates; this is due to frequent change in transmitting frequencies which yield more robust transmission links. Link error is very common in our domestic environments, where multi-path and fading are very common reasons for the degradation of wireless performance. Hence, we need to design a rate adaptation algorithm with minimum packet loss.

Rate Adaptation (RA) is important to maintain as per the channel condition. In 802.11n the system dynamically adjusts the Modulation Coding Scheme (MCS) based on the channel condition. Using single stream (SS) and double stream (DS), it thus offers a wide range of rate options, starting from 6.5Mbps and reaching 600Mbps at the maximum. RA can be categorized as sender side and receiver aware. This paper deals about the receiver aware RA, the sender will get the acknowledgement from the receiver and will change the MCS accordingly. The main reason for many people not using receiver aware RA, the device won't have the provision to get the information about the channel conditions. For our experiment, we are going to use Airpcap devices which have capacity to get channel information.

This study offers a receiver aware algorithm where the sender and receiver exchanges rate acknowledgement messages and change the modulation and coding schemes (MCS) accordingly. Received Signal Strength Indicator (RSSI) is used as an indicator for channel degradation in this study. In our signal Integrity based rate control algorithm, both MAC and PHY layers have joint impact on the performance of IEEE 802.11-based systems. As a result, the implementation evaluates the performance of open loop algorithm with the signal integrity based rate control algorithm proposed in the study. The algorithm developed in this study is an implemented using airpcap network interface which provides certain physical channel related parameters and pass it to the application. Using airpcap we avoided complexity of driver level implementation while achieving the same low-level access from application level software.

II. RELATED WORK

Automatic Rate Fallback (ARF) [1] is a well known rate adaptation algorithm which uses consecutive packets, consecutive frame losses, or timeout to switch different rates. Initially single packet frame is send at particular rate and it is lost then it will send the next packet at the same rate and start a timer. If the next packet is lost then the sender will send data at lower rate. The timer expires if the number of consecutive received packets reaches 10 and the station will send a probing frame at the next higher rate. However, ARF [1] can't rise up the rate easily due to the long consecutively successful packets. Adaptive ARF (AARF) [2] is the extension of ARF; it doubles the threshold of the consecutively successful received packets and the length of the timer while the probing frame is failed. The main idea of the AARF [2] is to reduce the trails of higher data rates, to get the better performance than ARF does it in stable channel conditions. The main problem in AARF [2] is that it lengthens the timer which leads to worst performance and also faces the same problem as of ARF [1].

Collision-Aware Rate Adaptation (CARA) [3] is to use RTS/CTS mechanism to effectively avoid collisions. CARA [3] assumes that RTS failures are caused by collision. When the first packet is lost, CARA [3] will enable RTS/CTS mechanism in the next frame. If the next transmission still fails, then the station will decrease the current data rate, otherwise, the station will continue the current rate and disable RTS/CTS mechanism. Although RTS/CTS mechanism prevents from packet loss from collision but it will take long time to adapt with different rates. Robust Rate Adaptation Algorithm (RRAA) [4] uses two loss ratios to determine rates and operates an adaptive RTS filter. RRAA [4] uses Maximum Loss Threshold and opportunistic rate Increase threshold to determine rates. According to the transmission time ratio of the adjacent rate, RRAA [4] is determines the window size for each rate. When the number of transmission count reaches to the window size of the current rate, RRAA [4] starts to do rate adaptation. If the ratio is larger than MTL threshold, the station decreases the rate in the next window and the window size will be reset by the new rate. If the ratio is smaller than ORI threshold, the station increases the rate in the next window. The rate remains the same in other conditions. In RRAA [4] nodes collect enough statistics to make correct decision in time before the timer expires again. It doesn't take RTS failure into account. It is hard to use RRAA [4] if we are planning to use many nodes, because it is difficult to get statistics at each node and to choose a proper rate.

Stochastic Automata Rate Adaptation (SARA) [5] algorithm uses the stochastic automata mechanism to adjust the probability of each rate. It set the equal probability for each rate in the beginning and chooses a data rate to transmit according to its probability. The mechanism of SARA [5] can be separate into three phases. a) Initial Phase: in the initial phase, every rate is chosen above N times (N is a parameter which can be changed by the user); b) trial Phase: In the trial phase, SARA [5] chooses a rate and computes its PSR; then it records the number of success and failure in the past N time interval; c) updating phase: SARA [5] will calculate the throughput of each rate according to PSR. Then it increases the probability for those which have the maximum throughputs and decreases the others. It chooses the rate with the highest probability to transmit the data and then SARA [5] goes back to the trail phase and updating phase again. Receiver Based Auto Rate (RBAR) [10] is for receivers to measure the channel quality using physical layer analysis of the Request-To-Send (RTS) message. Receiver than set the transmission rate for each packet according to the highest value allowed by the channel condition. In RBAR [10] lot of message is going to and fro which generally increases the traffic in the channel. Moreover RBAR [10] won't involve select any MCS for transmission.

Opportunistic Auto Rate (OAR) [11] exploits channel variability to increase the throughput of IEEE 802.11 ad-hoc networks. In particular, OAR [11] exploits the fact that at moderate velocities, channel coherence time is on the order of multiple packet times, such that when the channel quality is high, throughput improvement can be obtained by opportunistically sending multiple back-to-back packets at a higher rate. OAR [11] obtains a throughput gain as compared to RBAR and ARF and also ensures time-share fairness to ensure that users with perpetually bad channels obtain their fair share of time accessing the channel. In the paper [6], Performance Evaluation of rate adaptation algorithm in WLAN, whenever there is a packet loss over the network, they are changing the MCS to the minimum MCS so that the rate at which they are sending is minimized and the paper [6] won't consider the rate RSSI for processing. In 802.11n, for example we are trying to send data at some higher rate with high MCS; if packet is lost we have to change the MCS to 0. By the way we can reduce the rate to 6.5 mbps. Hence the time required to transmit the certain number of packets increases because the rate at which it is transmitted is very low.

III. Rate Adaptation by Signal Strength Monitor

RF interference is one of the problems in WLAN deployments; each RF stations will transmit packets when there is no station transmitting. If another station happens to be sending a packet, the other station will wait until the channel is free. Interference involves the presence of unwanted, interfering RF signals that disrupt normal system operations. Because of the 802.11 medium access protocol, an interfering RF signal of sufficient amplitude and frequency can appear as a bogus 802.11 station transmitting a packet. This causes legitimate 802.11 stations to wait for indefinite periods of time until the interfering signal goes away.

In general, if the RSSI is high, there will be very low interference. Similarly, if the RSSI level is low there is a very high interference in the channel. RSSI can be used by the receiver to internally to verify the amount of energy in the channel is below the certain threshold and receiver will send clear to send (CTS). When the sender receives CTS, it will transmit data. There is no standard relationship of any particular physical parameter r to RSSI reading. RSSI has been replaced with Received Channel Power Indicator (RCPI) in many parts. In this paper, we used RSSI range from 0-50 dbm. If it is 50 dbm, the signal level is very good and don't face any problem in sending at higher rates. On the other hand, if the RSSI value is 0 dbm, the signal level is very low and not able to send data in very high rate.

IV. PROPOSED ALGORITHM

In this thesis, we propose a robust and fast adaptive rate control algorithm. The main aim of the thesis is to achieve maximum throughput. Throughput is the number of error free information that is delivered to the upper layer of the receiver. Both Physical layer and MAC layer is mainly responsible for sending packets over the network; it is good to design a rate control algorithm which includes MAC and PHY layers.

MAC layer is responsible for maximum networking characters, such as rate, fragment number, sequence number, source address, destination address, etc. Receivers capture all the packets, but in order to check whether the packet belongs to that particular receiver can be verified using the destination address in the MAC header flag. PHY layer is responsible for transmitting data from sender to receiver. It includes header which deals mainly with the communication parameters such as RSSI, MIMO antennae, etc. In the algorithm explained below works well with packet loss due to collision but works will with packet loss due to signal degradation.

A. Algorithm Description:

Sender Algorithm:

1) Get the channel number, rate and MCS (static variable)
2) Search and select the available devices for transmission
3) Open the winpcap adapter for the airpcap device and set the device channel.
4) Set the link type as 802_11_PPI and include PPI header and 802.11 headers for transmission
5) Simultaneously start a thread which will receive data with new MCS send by the receiver.
6) If it gets MCS update then new MCS is updated and with the new MCS it will transmit data to the receiver.

Receiver Algorithm:

1) Open the Airpcap device and set the link type to 802_11_PPI
2) Read event is started and buffer is created for the receiving packets. Read the data of each packet.
3) Check the size of the received packet and the destination address of the packet. If it matches proceed else stop at this step
4) Record the number of packets received for the period of 30 sec and average the RSSI of all the received packets.
5) For every 30 sec calculate packet loss rate

$$\text{Original number of packets} = \frac{(rate * 1000000 * 30)}{\text{packet size} * 8}$$

$$\text{Packet loss rate} = \frac{(\text{original number of packets} - \text{Packets received})}{\text{original number of packets}} * 100\%$$

Reset packets received to 0.

6) If packet loss rate is greater than 15 % and Average RSSI is less than 10 dbm then change the MCS to 5 else keep the MCS as 12

Send the updated MCS to the transmitter with very low MCS (here its 0), which has packets reception rate of 100 %. Repeat the step 5 and 6 for every 30 sec

V. EXPERIMENTAL SETUP

A. PPI Header

When capturing live network data, it is often useful to collect out-of-band information and provide it along with in-band packet data. The traditional method of doing this is to prefix each packet with a meta-information header (often called a pseudo header). Current implementations include such information as 802.11 radio information, access server user IDs and point-to-point link direction. The main purpose of PPI header [9] is to provide 802.11n radio information and also to have other information. Each PPI header [9] field is a type-length-value triplet.

Packet Header	Field Header	Field Data	Field Header	Field Data

Figure 1: PPI Header Format

Each PPI header [9] may contain only a packet header with no field header or field data elements. The PPI packet header provides a version, indicator flag and header length. The current PPI version is set to 0. PPH flag is the flag that defines the behaviour of the header. PPH length indicates the length of the PPI header [9]. PPI field includes a type and length of a packet. The most common field types and length are mentioned below.

The main purpose of the PPI header [9] with MAC+PHY extension, over the 802.11n header is to include more physical layer parameters over the regular 802.11n header. The structure of the MAC + PHY PPI header [9] is furnished below.

Table 1: PPI Field Type and length

Type	**Length**	**Description**
2	20	802.11 common
3	12	802.11n MAC Extensions. Extended radio information
4	48	802.11n MAC+PHY extension. Extended radio information

Flags (4 bytes)	A-MPDU-ID (4 bytes)	Num-Delimiters (1 byte)	MCS (1 byte)	Num-Streams (1 byte)	RSSI-Combined	RSSI ANT0 ctrl
0	4	8	9	10	11	12 13
RSSI ANT1 ctrl	RSSI ANT2 ctrl	RSSI ANT3 ctrl	RSSI ANT0 EXT	RSSI ANT1 EXT	RSSI ANT2 EXT	RSSI ANT3 EXT
13	14	15	16	17	18	19 20
Ext-channel freq	Ext-channel flag	Dbm-ant0 signal	Dbm-ant0 noise	Dbm-ant1 signal	Dbm-ant1 noise	Dbm-ant2 signal
20	22	24	25	26	27	28 29
Dbm-ant2 noise	Dbm-ant3 noise	Dbm-ant3 signal	EVM0 (4bytes)	EVM1 (4bytes)	EVM2 (4bytes)	EVM3 (4bytes)
29	30	31	32	35	39	43 47

Figure 2: PPI header with MAC+PHY extension

In the entire header some values we need to fill at the sender end and some fields will be available at the receiver end, because most of the fields are based on the channel parameters like Received Signal Strength Indicator (RSSI), dbm's of antennae and Error Vector Magnitude's (EVM).

B. Floor Plan

The sleep time and the coding rate, stream, modulation scheme is found using the following equation.

$$Sleeptime(\mu sec) = \frac{(codingrate * numberofstreams * bitspercommunication * numberofsubcarrier)}{54}$$

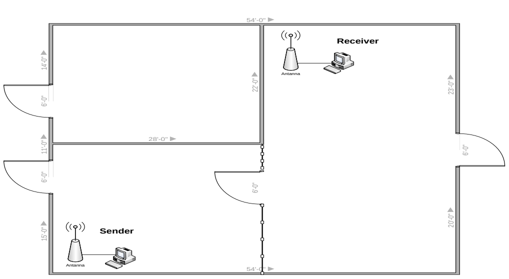

Figure 3: Floor Plan of transmitter and receiver with RSSI 45-35 dbm

Initially we started our experiment with the floor plan as shown in the figure 3, Packet Error Rate (PER), RSSI versus time when MCS 0 is used as the modulation and coding scheme. With MCS 0, a packet will be transmitted using BPSK, code rate of ½ and 1 spatial stream. In BPSK the number of bits transmitted per symbol is one, MCS 0 would not be suitable for very high data rate.

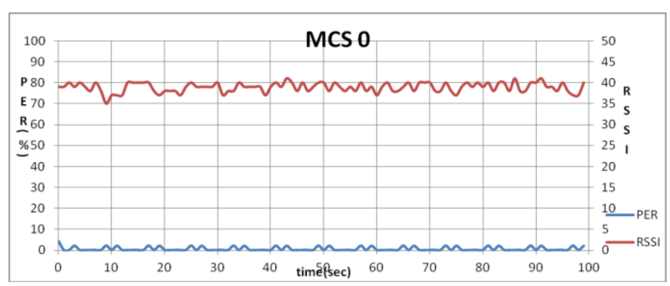

Figure 4: MCS 0 for RSSI 45-35 dbm

Figure 4 shows the packet error rate (PER) with MCS 0. With MCS 0 a packet will transmit with the modulation scheme of BPSK, coding rate of ½ and 1 spatial stream. As per equation (1) minimum sleep time is lower and the number of bits transmitted per symbol is also very less and this won't be suitable for very high data rate.

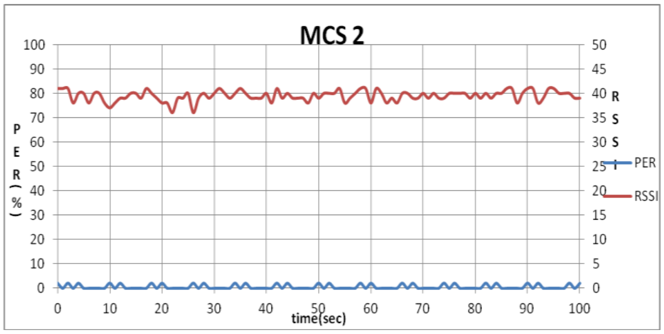

Figure 5: MCS 2 for RSSI 45-35 dbm

Figure 5 shows the packet error rate (PER), RSSI versus time for MCS 2. Using MCS 2, a packet will be transmitted with the modulation scheme of QPSK, coding rate of ¾ and 1 spatial stream. Figure 6 shows the packet error rate (PER), RSSI versus time for MCS 5. Using MCS 5, a packet will be transmitted with the modulation scheme of 64-QAM, coding rate of 2/3 and 1 spatial stream.

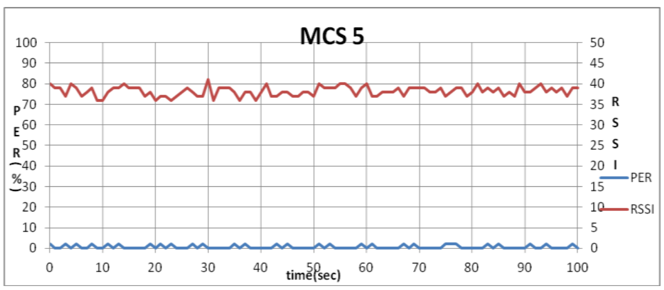

Figure 6: MCS 5 for RSSI 45-35

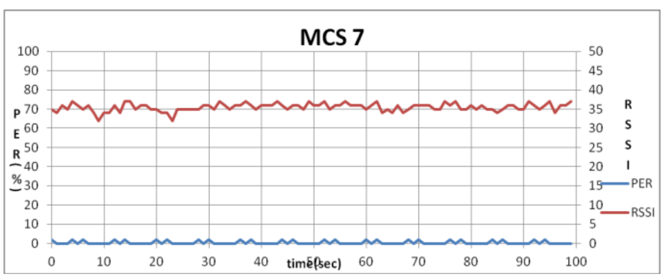

Figure 7: MCS 7 for RSSI 45-35

Figure 7 shows the packet error rate (PER), RSSI versus time for MCS 7. Using MCS 7, a packet will be transmitted with the modulation scheme of 64-QAM, coding rate of 5/6 and 1 spatial stream. In this case the bits transmitted per symbol are 6 and will be suitable for rate up to 72 Mbps.

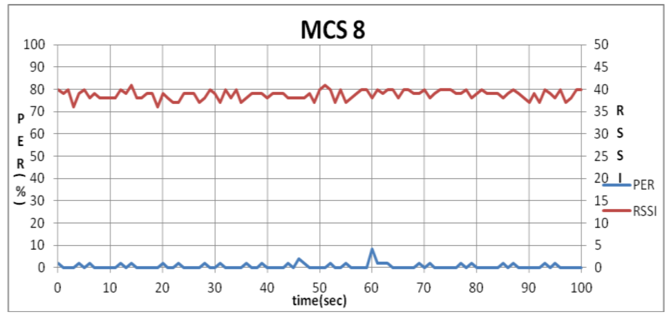

Figure 8: MCS 8 for RSSI 45-35

Figure 8 shows the packet error rate (PER), RSSI versus time for MCS 8. Using MCS 8, a packet will be transmitted with the modulation scheme of BPSK, coding rate of ½ and 2 spatial streams.

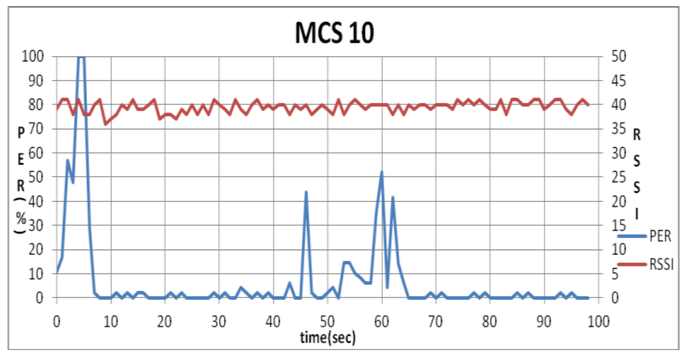

Figure 9: MCS 10 for RSSI 45-35

Figure 9 shows the packet error rate (PER) versus time for MCS 10. Using MCS 10, a packet will be transmitted with the modulation scheme of QPSK, coding rate of ¾ and 2 spatial streams. Maximum achievable rate using MCS 10 is 43 Mbps. There is a high packet loss between initial 0-8 seconds, because there some external factors which makes packets to loss. It is also noted that, when transmitting at higher MCS (MCS 12, MCS 15) there is packet loss for initial 8 seconds.

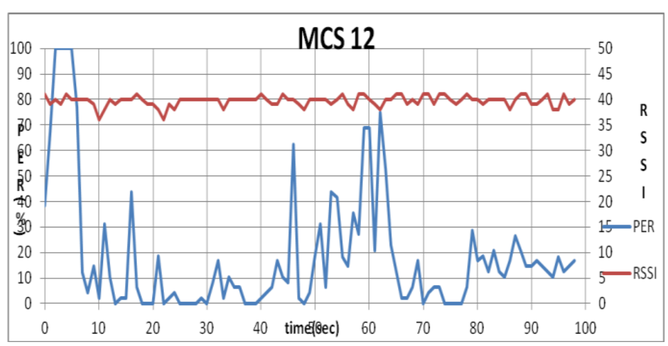

Figure 10: MCS 12 for RSSI 45-35

Figure 10 shows the packet error rate (PER), RSSI versus time for MCS 12. Using MCS 12, a packet will be transmitted with the modulation scheme of 16-QAM, coding rate of ¾ and 2 spatial streams.

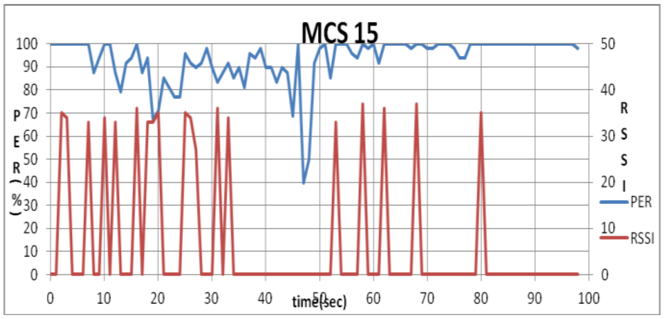

Figure 11: MCS 15 for RSSI 45-35

Figure 11 shows the packet error rate (PER), RSSI versus time for MCS 15. Using MCS 15, a packet will be transmitted with the modulation scheme of 64-QAM, coding rate of 5/6 and 2 spatial streams. Figure 4 to figure 11 show the packet error rate versus time for different MCS over the period of time. The results are more similar with all the MCS. In this scenario, most of the packets sent will reach the destination successfully and it is not a good idea to test the rate control algorithm in this situation. Figure 12 shows the packet error rate versus corresponding MCS.

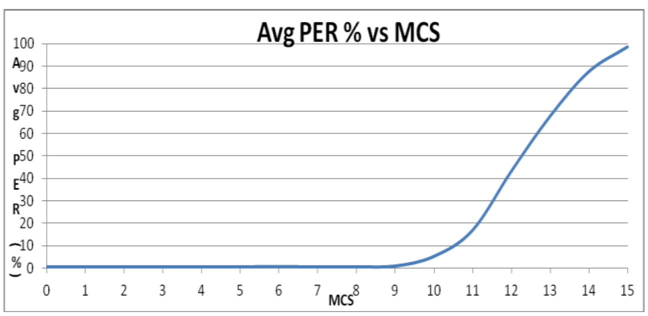

Figure 12: Average PER% vs MCS

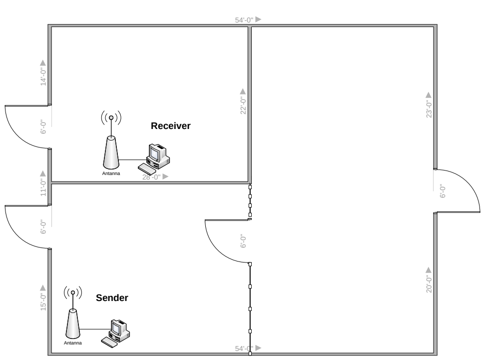

Figure 13: Floor Plan of transmitter and receiver with RSSI <10 dbm

Next we moved on to a position as shown in figure 13 where the RSSI is low (<10 dbm) and we captured the packet error rate graph with different MCS. This time we got better results than the previous set of experiments.

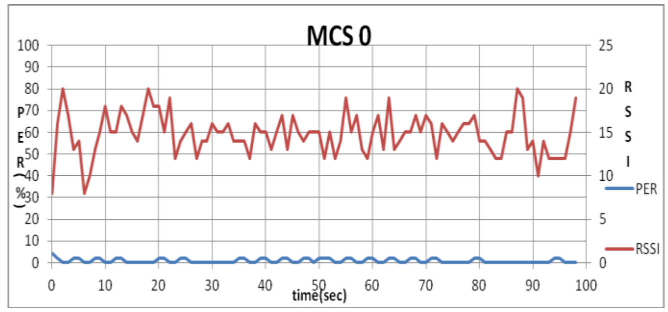

Figure 14: MCS 0 for RSSI <10

Figure 14 shows the packet error rate (PER), RSSI versus time for MCS 0 for floor plan 2. In Figure 14, it is clear that there is a very minimum packet loss when sending data in MCS 0. While checking the PER with different MCS in Floor plan 1, the packet error rate (PER) maintains throughout the experiment with minimum change in PER, but in Floor plan 2 packet error rate (PER) varies with increasing MCS and will be discussed in the present section.

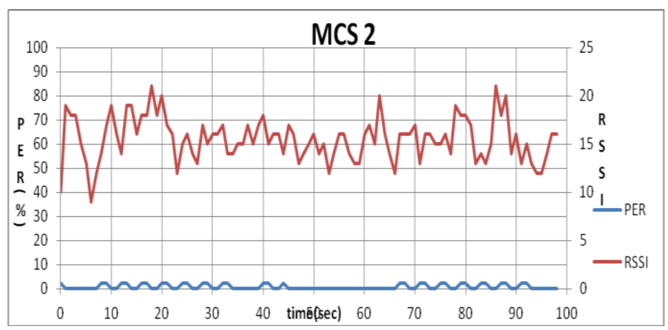

Figure 15: MCS 2 for RSSI <10

Figure 15 shows the packet error rate (PER), RSSI versus time for MCS 2 for floor plan 2. There would not be much difference in the pattern between Figure 14 and Figure 14 because MCS 0 transmits in BPSK and MCS 2 transmits in QPSK. Using this modulation scheme sender can transmit at rate of 21 mbps. Figure 16 shows the packet error rate (PER), RSSI versus time for MCS 4. With this modulation scheme, sender can transmit up to 28 Mbps.

Figure 17 shows the packet error rate (PER), RSSI versus time for MCS 5. With this MCS, sender can transmit up to 52 Mbps. Here, the PER is higher than the packets transmitted with MCS 4

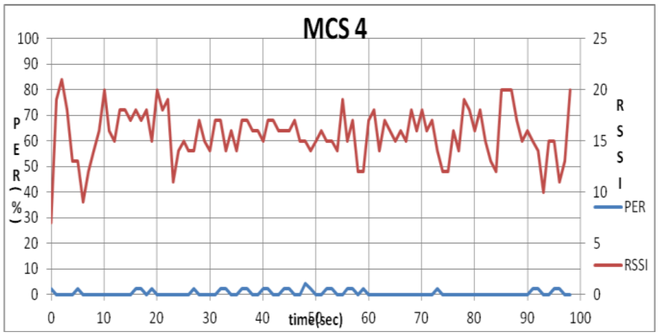

Figure 16: MCS 4 for RSSI <10

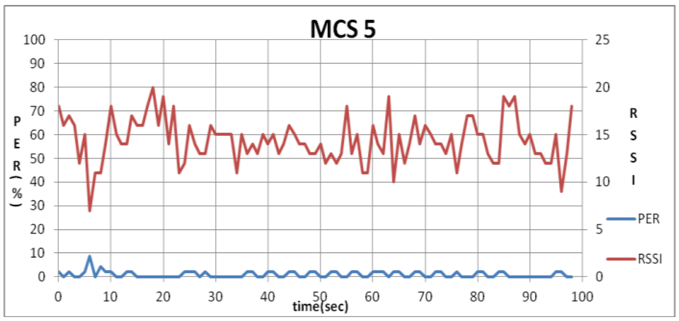

Figure 17: MCS 5 for RSSI <10

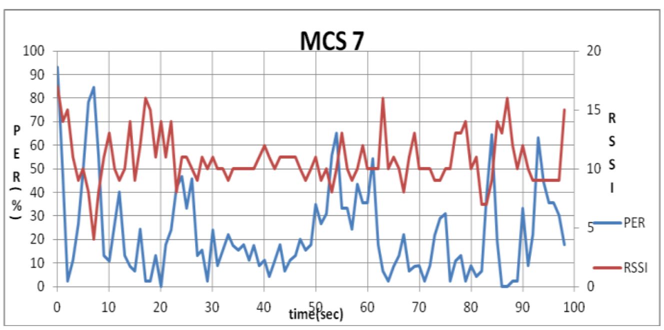

Figure 18: MCS 7 for RSSI <10

Figure 18 shows the packet error rate (PER), RSSI versus time for MCS 7. With this modulation scheme, sender can transmit up to 72 Mbps and suitable for high data rate transmission. Loss pattern can be easily differentiated in this case with the previous MCS transmission methods. With the current RSSI value (<10), the packet loss in the channel is very high and around 90% of the packets which transmitted in the channel was lost. The main advantage of having this modulation scheme is with very high good RSSI this MCS suits well.

Figure 19 shows the packet error rate (PER) with MCS 12. With MCS 12, a packet will transmit with the modulation scheme of 16-QAM, coding rate of ¾ and 2 spatial streams. Figure 19 explains there is much packet loss seen in MCS 12, when compared to MCS 10. Figure 20 shows the packet error rate (PER), RSSI versus time with MCS 8. This pattern is same as MCS 0 as shown in Figure 6.21. Using this modulation scheme, sender can transmit up to 14 Mbps and even with very low RSSI, MCS 8 can transmit with very minimum loss as shown in the Figure 9.

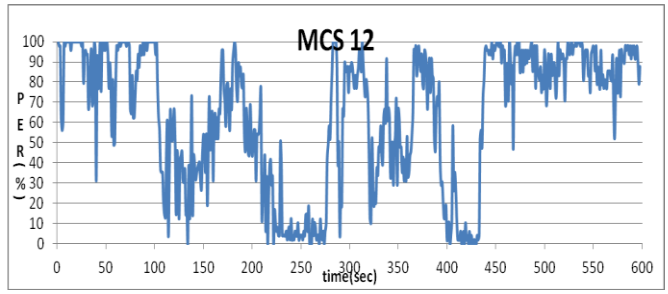

Figure 19: MCS 12 for RSSI <10 dbm

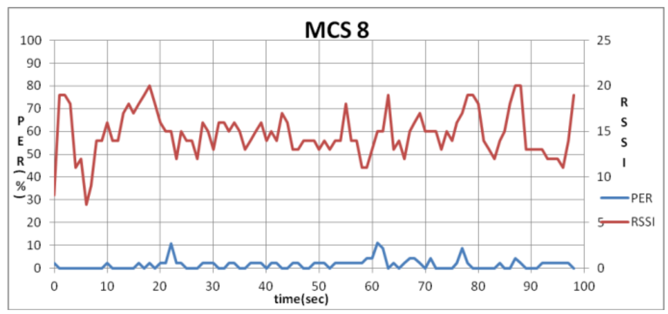

Figure 20: MCS 8 for RSSI <10

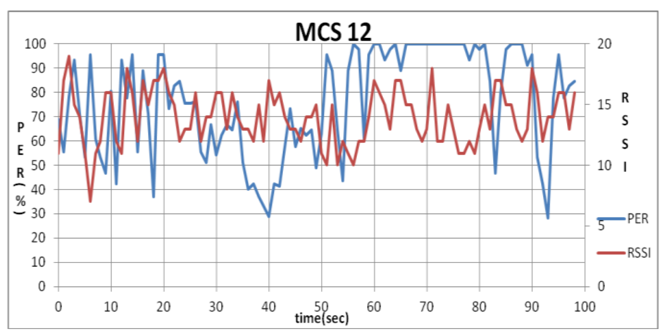

Figure 21: MCS 12 for RSSI <10

Figure 21 shows the packet error rate (PER), RSSI versus time for MCS 12. In this scenario, some of the packets are lost and some of them are successfully sent. Figure 22 shows the packet error rate (PER), RSSI versus time for MCS 15. In this case, almost all the packets have lost during the transmission.

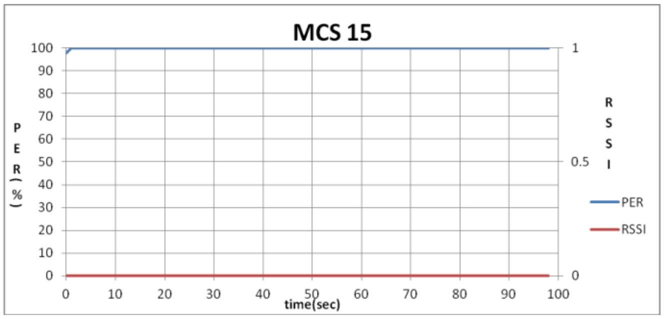

Figure 22: MCS 15 for RSSI <10

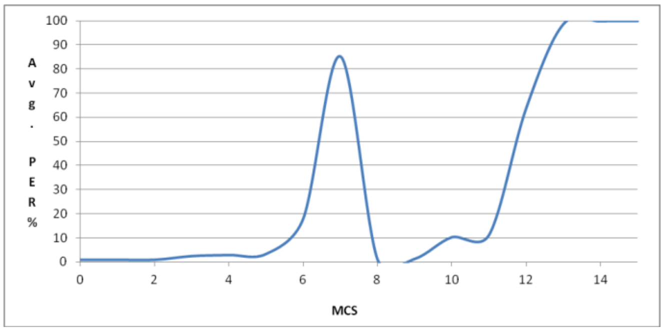

Figure 23: PER vs MCS

Figure 23 shows the MCS and their corresponding PER% for the RSSI < 10. In this scenario, there is a considerable increase in the packet error rate when the modulation scheme increases.

Table 2: Threshold PER for corresponding MCS

	MCS 13	MCS 12	MCS 10	MCS 8	MCS 5	MCS 4	MCS 1	MCS 0
Avg. PER (%)	72	51	26	2.37	3.78	0.95	0.96	0.807
TSH_{PER}	70	50	25	4	5	2	2	2

Based on the experiments conducted, threshold Packet Error Rate (PER) is calculated and shown in Table 2.

VI. Results

Once the position of the sender and receiver were decided, signal Integrity based rate Control algorithm is tested in with floor plan 2. Figure 24 explains the % of packet sent by signal Integrity based rate Control algorithm and Ministrel algorithm for a time period of 24 hours. It is important to conduct this experiment, because all the transmission are wireless and channel RSSI varies rapidly within short period. With signal Integrity based rate Control algorithm, very good throughput is achieved and transmitted 73% of packets over 50 mbps which is higher, when compared to the 41% transmitted over 50 mbps using Ministrel algorithm and is explained in Table 3.

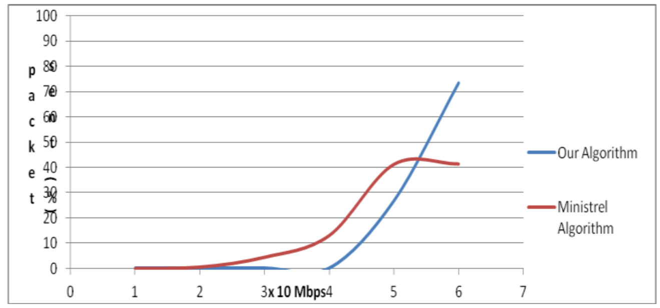

Figure 24: % of Packet Sent Vs Rate sent (mbps) for 24 hours

https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel Rates(Mbps)	0 to 10	10 to 20	20 to 30	30 to 40	40 to 50	> 50
Our Algorithm (%)	0	0	0	0	26.67	73.33
Ministrel Algorithm (%)	0	0.55	4.44	13.05	41.24	41.38

Table 3: % of transmission rates between Ministrel and signal Integrity based rate Control Algorithm

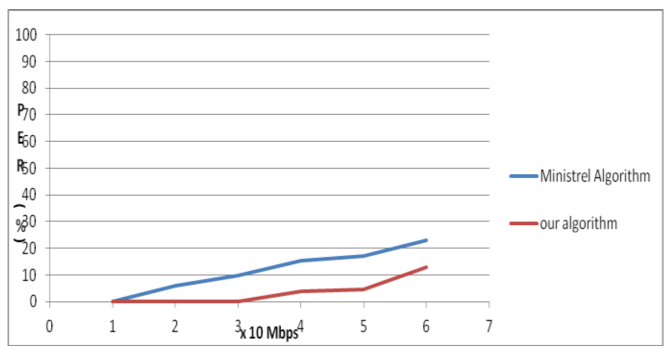

Figure 25: % PER of Ministrel Algorithm and our algorithm

Figure 25 shows the PER using Ministrel Algorithm and our algorithm. The average PER is 15% which is comparatively higher than the PER using our Signal Integrity based rate control algorithm which is 7.32%.

References

[1] A. Kamerman and L. Monteban, "Wavelan-ii: a high-performance wireless lan for the unlicensed band," Bell Labs Technical Journal, vol. 2,no. 3, pp. 118–133, 1997. ARF.

[2] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, (New York, NY, USA), pp. 126–134, ACM, 2004.

[3] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs," in INFOCOM 2006.

25th IEEE International Conference on Computer Communications. Proceedings, pp. 1–11, Apr. 2006.

[4] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking, (New York, NY, USA), pp. 146–157, ACM, 2006.

[5] T. Joshi, D. Ahuja, D. Singh, and D. Agrawal, "Sara: Stochastic automata rate adaptation for ieee 802.11 networks," Parallel and Distributed Systems, IEEE Transactions on, vol. 19, pp. 1579–1590, Nov. 2008.

[6] Shu-Ming Liu, Yu-Hong Lin, Wu-i Chou, Zan-Yu Chen, Tsung-Nan Lin, "Performance Evaluation of Rate Adaptation Algorithms in 802.11-based Mesh Networks," GLOBECOM Workshops, 2009 IEEE, IEEE Transactions on, vol. 19, pp. 1–5, Nov. 2009.

[7] Ioannis Pefkianakis, Yun Hu, Starsky H.Y. Wong, Hao Yang, Songwu Lu, "MIMO Rate Adaptation in 802.11n Wireless Networks", in MobiCom '10: Proceedings of the sixteenth annual international conference on Mobile computing and networking, (New York, NY, USA), pp. 110-121, 2010.

[8] https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel - Minstrel Algorithm

[9] http://www.cacetech.com/documents/PPI_Header_format_1.0.1.pdf - PPI Header Format

[10] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In Proceedings of ACM MOBICOM'01, Rome, Italy,2001.

[11] Liping Qian; Jinfang Zhou; Ying Qian; Kangsheng Chen; , "Improved Opportunistic Auto Rate Media Access Protocol for Multi-Rate Ad Hoc Networks," Wireless Communications, Networking and Mobile Computing, 2006. WiCOM 2006.International Conference on , vol., no., pp.1-4, 22-24 Sept. 2006S.

